

**CLAIMS:**

What is claimed is:

1. A method for booting a non-uniform-memory-access (NUMA) machine, comprising:
  - configuring a plurality of standalone, symmetrical multiprocessing systems to operate within a NUMA system;
  - assigning a NUMA identification to each of the multiprocessing systems, wherein each identification is unique; and
  - booting the multiprocessing systems in NUMA mode in one-pass, wherein memory coherency is established at the beginning of the execution of the system firmware.
- 15 2. The method according to claim 1, wherein the step of configuring the multiprocessing systems further comprises:
  - configuring and testing host processors and memory;
  - configuring and testing NUMA memory;
  - 20 loading firmware image into local memory and informing a hardware system console of the firmware version;
  - receiving a confirmation from the hardware system console that the firmware version is the same for all
  - 25 multiprocessing systems in the NUMA system;
  - configuring NUMA adapters to connect each multiprocessing system to the NUMA system and initializing all host processors; and
  - releasing all host processors to execute system
- 30 firmware.

3. The method according to claim 1, further comprising:  
selecting a nodal master processor within each  
multiprocessing system and designating all other  
processors within each system as nodal slave processors;  
5 and  
selecting a NUMA master processor from among the  
separate nodal master processors and designating all  
other nodal master processors as NUMA slave processors.
- 10 4. The method according to claim 3, further comprising:  
performing one-to-one handshaking between the nodal  
master processor and each nodal slave processor within a  
multiprocessing system, wherein the handshaking  
synchronizes the time base register of all nodal slave  
15 processors with the nodal time base source; and  
switching the nodal slave processors in each  
multiprocessing system to a hyper-visor environment in  
which the nodal slave processors become NUMA slave  
processors.  
20
5. The method according to claim 3, wherein the NUMA  
master processor:  
loads a single operating system into NUMA system  
memory;  
25 transfers control to the operating system; and  
runs operating system code which assigns all slave  
processors to a designated place within the NUMA  
operating system.  
30 6. A system for booting a non-uniform-memory-access  
(NUMA) machine, comprising:

- a plurality of hardware-configuring components which configure a plurality of standalone, symmetrical multiprocessing systems to operate within a NUMA system;
  - 5 an identification component which assigns a NUMA identification to each of the multiprocessing systems, wherein each identification is unique; and
  - a booting mechanism which boots the multiprocessing systems in NUMA mode in one-pass, wherein memory coherency is established at the beginning of the
  - 10 execution of the system firmware.
7. The system according to claim 6, wherein each hardware-configuring component further comprises:
- a host testing component which configures and tests host processors and memory;
  - 15 a NUMA testing component which configures and tests NUMA memory;
  - a software loading mechanism which loads a firmware image into local memory and informs a hardware system console of the firmware version;
  - 20 a receiving component which receives a confirmation from the hardware system console that the firmware version is the same for all multiprocessing systems in the NUMA system;
  - 25 an adapter-configuring component which configures NUMA adapters to connect each multiprocessing system to the NUMA system and initializing all host processors; and
  - a releasing mechanism which releases all host processors to execute system firmware.

8. The system according to claim 6, further comprising:  
a plurality of nodal selection mechanisms which  
select a nodal master processor within each  
multiprocessing system and designate all other processors  
5 within each system as nodal slave processors; and

a NUMA selection mechanism which selects a NUMA  
master processor from among the separate nodal master  
processors and designates all other nodal master  
processors as NUMA slave processors.

10

9. The system according to claim 8, further comprising:

a handshaking mechanism which performs one-to-one  
handshaking between the nodal master processor and each  
nodal slave processor within a multiprocessing system,  
15 wherein the handshaking synchronizes the time base  
register of all nodal slave processors with the nodal  
time base source; and

a switching mechanism which switches the nodal slave  
processors in each multiprocessing system to a  
20 hyper-visor environment in which the nodal slave  
processors become NUMA slave processors.

10. The system according to claim 8, wherein the NUMA  
master processor:

25 loads a single operating system into NUMA system  
memory;

transfers control to the operating system; and  
runs operating system code which assigns all slave  
processors to a designated place within the NUMA  
30 operating system.

11. A computer program product in a computer readable medium for use in a data processing system, for booting a non-uniform-memory-access (NUMA) machine, the computer program product comprising:

- 5       instructions for configuring a plurality of  
standalone, symmetrical multiprocessing systems to  
operate within a NUMA system;

10      instructions for assigning a NUMA identification to  
each of the multiprocessing systems, wherein each  
identification is unique; and

15      instructions for booting the multiprocessing systems  
in NUMA mode in one-pass, wherein memory coherency is  
established at the beginning of the execution of the  
system firmware.

5           12. The computer program product according to claim 11,  
wherein the step of configuring the multiprocessing  
systems further comprises:

0        instructions for configuring and testing host  
processors and memory;

0        instructions for configuring and testing NUMA  
memory;

5        instructions for loading firmware image into local  
memory and informing a hardware system console of the  
firmware version;

0        instructions for receiving a confirmation from the  
hardware system console that the firmware version is the  
same for all multiprocessing systems in the NUMA system;

0        instructions for configuring NUMA adapters to  
connect each multiprocessing system to the NUMA system  
and initializing all host processors; and

instructions for releasing all host processors to execute system firmware.

13. The computer program product according to claim 11,  
5 further comprising:

instructions for selecting a nodal master processor within each multiprocessing system and designating all other processors within each system as nodal slave processors; and

10 instructions for selecting a NUMA master processor from among the separate nodal master processors and designating all other nodal master processors as NUMA slave processors.

15 14. The computer program product according to claim 13,  
further comprising:

instructions for performing one-to-one handshaking between the nodal master processor and each nodal slave processor within a multiprocessing system, wherein the  
20 handshaking synchronizes the time base register of all nodal slave processors with the nodal time base source;  
and

instructions for switching the nodal slave processors in each multiprocessing system to a  
25 hyper-visor environment in which the nodal slave processors become NUMA slave processors.

15. The computer program product according to claim 13,  
wherein the NUMA master processor:

30 loads a single operating system into NUMA system memory;

transfers control to the operating system; and runs operating system code which assigns all slave processors to a designated place within the NUMA operating system.

5

16. A system for booting a non-uniform-memory-access (NUMA) machine, comprising:

a plurality of hardware-configuring components which configure a plurality of standalone, symmetrical

10 multiprocessing systems to operate within a NUMA system;

a host testing component which configures and tests host processors and memory;

a NUMA testing component which configures and tests NUMA memory;

15 a software loading mechanism which loads a firmware image into local memory and informs a hardware system console of the firmware version;

a receiving component which receives a confirmation from the hardware system console that the firmware

20 version is the same for all multiprocessing systems in the NUMA system;

an adapter-configuring component which configures NUMA adapters to connect each multiprocessing system to the NUMA system and initializing all host processors;

25 a releasing mechanism which releases all host processors to execute system firmware;

an identification component which assigns a NUMA identification to each of the multiprocessing systems, wherein each identification is unique;

30 a plurality of nodal selection mechanisms which select a nodal master processor within each

multiprocessing system and designate all other processors within each system as nodal slave processors;

a NUMA selection mechanism which selects a NUMA master processor from among the separate nodal master

- 5 processors and designates all other nodal master processors as NUMA slave processors;

a switching mechanism which switches the nodal slave processors in each multiprocessing system to a hyper-visor environment in which the nodal slave

- 10 processors become NUMA slave processors; and

a booting mechanism which boots the multiprocessing systems in NUMA mode in one-pass, wherein memory coherency is established at the beginning of the execution of the system firmware.

PCT/US2019/033036